

## Hacking events: project development practices and technology use at hackathons

Article (Published Version)

Richterich, Annika (2019) Hacking events: project development practices and technology use at hackathons. *Convergence*, 25 (5-6). pp. 1000-1026. ISSN 1354-8565

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/86223/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

### **Copyright and reuse:**

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

# Hacking events: Project development practices and technology use at hackathons

**Annika Richterich**

Maastricht University, The Netherlands

Convergence: The International  
Journal of Research into  
New Media Technologies  
2019, Vol. 25(5-6) 1000–1026  
© The Author(s) 2017



Article reuse guidelines:  
[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)  
DOI: 10.1177/1354856517709405  
[journals.sagepub.com/home/con](https://journals.sagepub.com/home/con)



## Abstract

Hackathons are techno-creative events during which participants get together in a physical location. They may be hosted by civic communities, corporations or public institutions. Working individually or in teams, usually for several days, participants develop projects such as hardware or software prototypes. Based on a digital ethnography of two events in the Netherlands and Denmark, this article investigates project development practices at hackathons. In particular, it analyses how participants organized their project work and which technologies were used in support of their creative endeavours. Hackathons are increasingly competitive rather than collaborative events, involving time pressure, inducements such as prizes, and requiring efficient skills utilization. I argue that this facilitates the following tendencies: Firstly, strategic effort is put into final presentations. Projects need to be convincingly presented, and persuasively pitching an idea becomes crucial. Secondly, there is only limited time for personal learning, since participants' existing skills need to be efficiently applied if a team wants to stay competitive. This encourages division of labour within groups: a tendency which seems especially problematic given that IT skills biases are often expressed in terms of gender. Thirdly, participants are more inclined to use technologies that are proprietary but appear 'open enough'. In light of this observation and by drawing on the concept of *technology as resource and opportunity*, I discuss the techno-political implications of utilized technologies. With this analysis, I aim at contributing to the critical debate on hackathons as productive but likewise ideologically significant fields of 'hacking cultures'.

## Keywords

Digital ethnography, hackathon, hacking cultures, IT skills, open source, techno-politics, technology and gender

---

## Corresponding author:

Annika Richterich, Faculty of Arts and Social Sciences, Maastricht University, Grote Gracht 90/92, 6211 SZ Maastricht, The Netherlands.

Email: [a.richterich@maastrichtuniversity.nl](mailto:a.richterich@maastrichtuniversity.nl)

## Introduction

During the OpenBSD hackathon in 1999, a group of volunteer programmers got together at a venue in Calgary, Canada. For 1 week, they joined forces and worked on an open-source operating system (see OpenBSD, n.d.). The event was based on practices of collaborative software development that were already common in developer communities back then. However, for the first time, the term ‘hackathon’ was used to describe such a meeting. What started in the context of an open-source project was soon discovered by companies. Later in the year 1999, Sun Microsystems hosted a hackathon dedicated to the digital assistant Palm-V. By now, all kinds of hac[king-mar]athons tap into the innovation capacity of collaborative programming and prototyping (see e.g. Goggins et al., 2014; Kaltenbrunner and Echtler, 2014; Quilez and Pol, 2014).

During hackathons, participants meet in physical public or private spaces in order to create software and/or hardware prototypes. They receive catering and might even stay overnight, since these events usually take several days. Nowadays, there are still hackathons organized by non-profit communities or foundations aimed at the development of technologies for the common good. Such rather socially oriented objectives apply for instance to the 2013 GNU’s hackathon for the 30th anniversary of the free software operation system.<sup>1</sup> At the same time, such events are increasingly common in commercial settings, for example hosted by companies looking for a technological solution to an in-house problem or trying to trigger employees’ creativity (see Rosell et al., 2014). An example are the regular Facebook hackathons, organized since 2004, with which the company states to give ‘[...] our employees the opportunity to try out new ideas and collaborate with other people in a fun environment’.<sup>2</sup> This popularization of hackathons has created discrepancies and frictions, as pointed out by Briscoe and Mulligan (2014: 4), ‘[M]any hackathons have benefited from professional organisation, utilising corporate sponsorships and investor-participation. However, this has occurred in some instances with tensions due to the socially orientated innovation often prevalent in earlier hackathons’.

The authors describe how monetary benefits have been perceived as contradictory to the values relevant to hackathons which emphasized intrinsic, philanthropic motivations. While Briscoe and Mulligan accurately highlight the existence of tensions between different types of hackathons, it remains to be explored how competitive (monetary) inducements and the involvement of sponsors may influence hackathons ‘on the ground’. Therefore, this article will empirically explore how project development practices at hackathons have developed in increasingly professionalized contexts.

Trainer et al. (2016: 8) provide insights into the socio-technical trade-offs of hackathons, suggesting that during their involvement participants negotiate ‘[...] between advancing technical work and building social ties’. Critical reflections on this issue have also been voiced by hackathon participants themselves. In a *Medium* post titled ‘Selling out and the Death of Hacker Culture’ (2015), former hackathon enthusiast Rodney Folz reflects on the changed experience of joining such events. The first sentences refer to early hackathons in which he participated:

We worked on whatever we felt like, asked each other about our projects, helped debug things when we got stuck. Everyone was there for the joy of building things with technology. Nobody cared about anything else. Later on, the CSUA [*Computer Science Undergraduate Association* at UC Berkeley] had a hackathon. With sponsors. Yahoo! was there, and some other companies whose names now escape me. [...]; there were now first, second, and third places, with associated prizes. I remember these prizes being alluring – people were less willing to talk about what they were building, less willing to

help others debug, because that was time they could spend building their hack. And if they had the best hack, they could beat everyone else and take first place. Yahoo! might even hire them. (Folz, 2015)

Of course, this comment reflects an individual perspective on the described changes and can only be seen as one possible scenario. However, it is the illustrated interplay between the organizational context and participants' interactions which is likewise of interest for this article. In addition, I will argue that not only participants' interactions and the organizational setup are co-constitutive, but also their use of certain technologies. Therefore, my main research question is how do the organizational conditions of hackathons interact with project development practices and participants' use of certain technologies?

### *Approach and main aims*

Methodologically, this article is based on digital ethnography, involving participant as well as non-participant observations and interviews during two hackathons. I will elaborate on this approach further below. The analysis presents insights from a *Science-Hack-Day* (SHD) in Eindhoven (the Netherlands) and *Hack4DK*, a museum hackathon in Copenhagen (Denmark). By drawing on observations from these field sites, interaction and conversations with participants, I will argue that competitive incentives and time pressure at hackathons encourage the following tendencies:

1. Strategic effort is put into persuasive presentations; the idea needs to be convincing, but the technological implementation not necessarily completely functional.
2. There is only limited time for personal learning and development of new skills, since participants' expertise needs to be efficiently applied. This is particularly problematic, since different skills (levels) at hackathons are often expressed in terms of gender – also due to a general gender bias in IT domains.
3. Technology choices and their utilization are approached pragmatically: Participants are inclined to use technologies that are proprietary rather than open source but appear 'open enough'.

In light of the latter argument, it is important to stress that utilized technologies may enforce technological (in-)dependencies and signify particular values regarding production, technological transparency and information freedom. The following section will therefore first reflect on the significance of techno-political implications more generally. It will do so by drawing particularly on Postigo's concept of *technology as resource and opportunity* (2012). Secondly, I will specifically illustrate the techno-political implications of free and open-source software and hardware in contrast to proprietary tools. This section is relevant since these implications are also reflected in the technology choices made by hackathon participants. Thirdly, in order to contextualize the hackathon examples analysed in this article, I will provide a comparison of early hackathons. In the fourth section, I will elaborate on my research approach: digital ethnography. Subsequently, I will present the main insights from my field research at selected hackathons, relating these observations to existing debates on utilized technologies and the increasing operational openness of proprietary tools. The conclusion finally highlights my main arguments and their practical relevance.

With this analysis, I aim at contributing to the critical debate on hackathons as productive, but likewise ideologically significant fields of hacking cultures. Authors such as Irani (2015: 801) have

already thoroughly analysed how hackathons encourage and produce ‘entrepreneurial subjects’: the author illustrates how a hackathon in India ‘[...] rehearses an entrepreneurial citizenship celebrated in transnational cultures that orient toward Silicon Valley for models of social change’. Similarly, Gregg (2015) suggests that ‘[h]ackathons enable aspiring professionals to demonstrate technical skills and immaterial qualities of employability’. These authors critically analyse selected hacking events and their implications for created subject positions. However, they pay less attention to the development that hackathons are no longer platforms where only equally qualified professionals or students meet. Neither are the motivations to join a hackathon always to be found in networking with peers, for instance. Instead, the popularization of hackathons has also facilitated a moderately increased heterogeneity of participants. While one may argue – as Nandi and Mandernach (2016) do – that hackathons *could* function as ‘informal learning platforms’, I question the conditions under which this is supposed to be the case. I particularly aim at addressing the dynamics emerging from related skills differences (which may be reflected in terms of gender) in relation to organizational conditions as well as used technologies.

## Technology as resource and opportunity

Participants’ technology choices at hackathons seem particularly relevant since the utilized tools are by no means neutral but have techno-political implications. Postigo suggests (2012: 177) that technology may function as *resource* and *opportunity*: ‘[T]hinking of technology as a resource is to think of it not only as a functional resource, but also as a linguistic resource. Technology speaks a certain kind of language about cultural production’. In the context of this article, this is crucial, since technology used at hackathons is staged as enabler of creative practices and digital production. According to Postigo, technology moreover acts as opportunity. It allows ‘for a particular relationship between consumers and content’ (Postigo, 2012: 177). This argument is also reflected in Coleman’s (2004: 507) examination of the ‘informal politics’ of FOSS. Coleman describes this technology as a ‘catalyst by which to rethink the assumptions of intellectual property rights through its use and inversion’ (Coleman, 2004: 508; see also Wark, 2013).<sup>3</sup>

As Postigo (2012: 8) points out with regard to, for instance, technologies circumventing copy protection: *the material presence of such technologies realizes the world they seek* (original emphasis). Technologies may be built with certain purposes and normative implications in mind. Of course, this is not a simple, techno-deterministic process: such purposes and implications have been frequently breached. However, they co-constitute common practices which are in turn likely to promote certain aims and values. Postigo applies this to technologies empowering the digital rights movement, but this rationale likewise accounts for proprietary soft- and hardware. These technologies serve as opportunity for the industry, by legally and technically enforcing limited interaction possibilities and usage conditions. Seeing that ‘design practices are explicitly political’ and that ‘technologies are explicitly meaningful’ (Postigo, 2012: 8), the outlined ideas also relate to prototyping more generally (see also Suchman et al., 2002) and to development practices at hackathons particularly. Since the use of certain technologies during hackathons may have techno-political implications – that is, leading to technological or economic (in-)dependence and information (non-)transparency – the following section will briefly discuss the core assumptions of free and open-source proponents in contrast to proprietary software.

## Hacking cultures: FOSS/free and open-source hardware

Hacking cultures are strongly related to (often even identical with) communities concerned with the development of Free and open-source software (FOSS) (Coleman, 2004, 2009, 2012; Himanen, 2010; Jordan, 2008, 2016; Lin, 2007; Nissenbaum, 2004; Söderberg, 2008). The historical evolution of the open-source movement (OSM) '[...] can be traced back to the "hacker culture" that created Unix, Linux, and parts of the Internet infrastructure' (Zhao and Elbaum, 2003, 66; see also Bergquist et al., 2011; Hippel and Krogh, 2003; Kely, 2008). Open source as production and development model ensures that anyone may have unrestricted access to a product's structure, blueprint, and design. The open-source model can be applied to a wide range of sectors (see Kogut and Metiu, 2001; McLellan et al., 2012; O'Boyle, 2011). In the context of software, open source implies that the source code of a programme is accessible to the public, can be used and modified.<sup>4</sup> The possibility of its commercial reuse is a condition for being acknowledged as open-source software.

The notion of open-source hardware (OSH) transfers the open-source software concept to material technology and components (on open-source appropriate technology, see Pearce, 2012; Powell, 2012). It refers to hardware which is provided with detailed information allowing for reproduction and further development of a material device. Originally, the emergence of OSH is related to the Open Design Foundation.<sup>5</sup> The concept extends the idea of detailed documentation of programming processes to the documentation of hardware development. The aim is to allow for hardware reproduction from scratch. Hardware design may be documented by mechanical drawings, providing the integrated (graphic) circuit layout. In this sense, OSH is a systematic continuation of the open-source software ideas. The licenses used for OSH are often derived from OSS licences; they are however closer connected to patent law rather than copyright law.

### Hacker values

Historically, the OSM is a spin-off of the free software movement (FSM). Both share the basic assumption that access to the source code is 'beneficial' (see also Coleman, 2004; Söderberg, 2008). However, they do so for different reasons. The FSM stresses altruistic motives; the OSM highlights practical advantages: '[F]ree software bases its activity on the argument that sharing code is a moral obligation and open source bases its activity on a pragmatic argument that sharing code produces better software' (Yeats, 2007: 13). The FSM suggests that the freedom to access, understand and modify software is a societal right which is infringed by proprietary (closed-source) software. In this sense, proprietary software inhibits citizens' individual and communal possibilities for learning, sharing and intellectual development (Stallman, 2007). Despite these differences, in the sense of Postigo's notion of *technology as resource*, FOSS likewise advocate values such as (code as) free speech (Coleman, 2004: 511ff.) and information transparency. In terms of *technology as opportunity*, they oppose mere proprietary approaches to intellectual property rights and ensuing barriers to digital creativity.

Going back to the hacker ethic, depicted by Levy in 1984, '[h]ackers believe that essential lessons can be learned about the systems – about the world – from taking things apart, seeing how they work, and using this knowledge to create new and even more interesting things' (Levy, 1996 [1984]). The author explains the need for non-proprietary, open systems, 'If you don't have access to the information you need to improve things, how can you fix them? [...] The best way to promote this free exchange of information is to have an open system' (Levy, 1996 [1984]). Yet, as Coleman and Golub (2008: 255) remark with reference to Elias Ladopoulos (Acid Phreak, 1990),

there is no *universal* hacker ethic, but rather ethical diversity among hackers. The authors argue that '[...] hacker morality in fact exists as multiple, overlapping genres that converge with broader prevailing political and cultural processes, such as those of liberalism' (Coleman and Golub, 2008: 256; see also Coleman, 2012, 2014). Similarly, Raymond stresses in a chapter on 'the varieties of hacker ideology':

[T]he FSF [Free Software Foundation] was never the only game in town. There was always a quieter, less confrontational and more market-friendly strain in the hacker culture. The pragmatists were loyal not so much to an ideology as to a group of engineering traditions founded on early open-source efforts which predated the FSF. (Raymond, 2005)

This article likewise acknowledges that varying values and aims may guide participants' decisions during hackathons. I will illustrate though that technology utilization during competitive events is often mainly guided by pragmatic approaches as described above. This tendency is described as 'hacker pragmatism'. The realization of such a pragmatic perspective is nothing new (see also Yeats, 2007). It is rather an inherent aspect of hacker culture manifestations such as the OSM. However, my argument highlights the ramifications of such a perspective and illustrates that it can be a crucial factor of influence for the technologies utilized during hackathons.

## Early hackathons and topical literature

While the term 'hacking' has been used in developer contexts since the 1960s, it was only in the late 1990s that the term hackathon emerged. It is beyond the scope of this article to provide an overview of the different historical (communal, corporate or institutional) contexts and related motives for which this notion was appropriated. It would likewise be interesting to investigate how the term simply labelled and reframed long-existing types of developer events. Drawing on Foucault, Jordan highlights with regard to hacking culture that one should avoid writing 'history from the point of view of either the "now", seeing past events as significant only if they lead to our present state, or in a closely related way, the "victor"' (2016: 2). While this is likewise true for hackathons, a 'genealogy' of these hacking events would require more than a sub-section in this article. What I am therefore presenting in the following sections is by no means a complete overview of hackathons' various manifestations and phases. Instead, the following reflections are merely meant to provide insights into developments that are particularly relevant for understanding the background and tensions regarding the investigated hackathons.

The term itself was originally used as metaphor for early firewall testing, 'Too often a firewall test is viewed as a kind of "hackathon". Organizations may authorize someone to conduct the testing, but this person may "disappear behind a black curtain"' (Moyer and Schultz, 1996: 12). With regard to activities involving collaborative programming and software development, hackathon was first prominently used in 1999. According to OpenBSD, the term was coined for their event taking place on 4–6 June 1999, 'In the months leading up to this, either Theo or Niels Provos had coined this new word "hackathon"' (OpenBSD, n.d.). Only a few weeks later, in September 1999, another pioneer hackathon took place in San Francisco. It was part of a JavaOne conference (Aviram, 1999) and called 'The Hackathon'. This illustrates how the coining of the term was subject to claims and negotiations.

In the Sun hackathon, participants were framed as competitors. They were asked to write a programme in Java for the personal digital assistant Palm-V (released in February 1999) and were competing for monetary prizes. In contrast, the idea behind the OpenBSD hackathon was that

developers would work collaboratively towards a common goal. The latter event was open to invited participants only. While the Sun hackathon took place in a commercial context, the OpenBSD hackathon was part of a community insisting on uncompromised open-source code and software licensing. These early hackathons already illustrate crucial differences between contemporary events: Some hackathons are conceptualized as collaborative, temporary think tanks working on ideas that are supposed to contribute to the common good. Others are used as fruitful breeding ground for commercially promising prototypes: 'During the 2000s hackathons became significantly more widespread, and began to be increasingly viewed by companies and venture capitalists as an approach to quickly develop new software technologies, and to locate new areas for innovation and funding' (Briscoe and Mulligan, 2014: 4). In such hackathons, subgroups or individual participants usually compete with each other rather than collaborating on a common project. Meanwhile, hackathons are indeed frequently suggested as problem-solving tools for various issues (Aungst, 2015; Johnson and Robinson, 2014; Munro, 2015). This perspective seems however rather simplistic and has been criticized (Irani, 2015; Porway, 2013).

Briscoe and Mulligan suggest differentiating between tech-centric and focus-centric hackathons. 'Tech-centric' refers to events focusing on software development using a specific technology or application; 'focus-centric' hackathons require participants to address societal issues or business objectives (see Briscoe and Mulligan, 2014: 5ff.). Regarding the projects developed at these events and the kinds of interaction taking place, one needs to take into account that many hackathons are predominantly frequented by male participants. A more extensive investigation focused on gendered innovation and hackathons as often male dominated domains – related to the programming communities they address (Adam, 2003; Jordan and Taylor, 2004: 116ff.) – would therefore be insightful. I will come back to this important point, but it is not the main focus of this article.

## Digital ethnography as research approach

This study is based on the digital ethnography approach suggested by Pink et al. (2016; see also Underberg and Zorn, 2013). The authors introduce this qualitative approach as way to explore '[...] the digital-material environments that we inhabit and how human activity and the environments in which it takes place are co-constitutive' (Pink et al., 2016: 152). Earlier methodological deliberations suggested for ethnographic research on digital practices, such as virtual ethnography (Gatson and Zwerink, 2004; Hine, 2000, 2008; Dominguez, Beaulieu and Estalella, 2007) or Internet ethnography (Miller and Slater, 2001), already referred to the relevance of physical practices and spaces for understanding digital/online activities. Pink et al. however particularly highlight the necessity to methodologically traverse between online and offline – just as the subjects of digital media research do. The authors emphasize that digital ethnography is in fact *non-digital centric* (see also Couldry, 2012). The approach does not focus on intangible practices and digital spaces but instead addresses how the '[...] digital has become part of the material, sensory and social worlds we inhabit' (Pink et al., 2016: 153). It is one of the main aims of this article not only to illustrate how digital technologies are selected and utilized during digital-material hackathon events but to show the co-constitutive interplay between social, technological and contextual factors. Therefore, my investigation draws on the research approach outlined in Pink et al. (2016) by exploring how participants alternate between and merge digital and material interaction.

In particular, this article was inspired by the chapter 'Researching events' (Pink et al., 2016: 147ff) aimed at '[...] understanding the digital, material, affective, and social relations of events'



(p. 152). As the authors stress, digital ethnography is not a static template which may be applied but rather needs to be individually developed. I have structured my analysis according to the three initially mentioned themes that I encountered during my observation and that highlight the connection between digital practices and physical conditions:

1. relevance and implications of persuasive presentations,
2. efficient skills utilizations and learning limitations and
3. pragmatic technology choices and uses.

I focus on these aspects since they were recurring, dominant themes demonstrating the interplay between organizational decisions and participants' project development practices.<sup>6</sup>

My insights are derived from field research at two hackathons: a SHD in Eindhoven (30–31 August 2014) and *Hack4DK* in Copenhagen (26–28 September 2014). The results presented in this article are based on (informal) interviews<sup>7</sup> and (non-)participant observations of activities on-site and online (e.g. on Twitter or Github). For my analysis, I made field notes of my observations, recorded and/or documented the interviews as well as the digital material.

During the SHD, I did participant observation via having an active role in the hackathon. I was part of a group of seven people with whom I experienced the development process. Initially, 14 groups were working on competing projects during this hackathon, but not all groups stayed until the end of the event (eventually 12 projects were presented). I did informal interviews with 21 participants, 2 of them were female, and recorded interviews with the main organizer before and during this hackathon. For *Hack4DK*, I chose a different approach: I did not participate in a particular project but joined with a visitor ticket. I attended the official introduction and presentations as part of the general audience. In addition, I approached the participants who were working in groups or individually, asking whether they would be willing to tell me about their projects and if I could watch their work. I talked to 18 participants, 4 of them were female. I joined six groups repeatedly during the event and at different development stages. Overall, eight projects were presented at the end of *Hack4DK* (seven, topically relevant projects are mentioned on <http://hack4.dk/projects-2014>). The aforementioned numbers were at times difficult to determine, since in many cases I would initially speak to one or two participants sitting at a table, but more group members might join in or comment on our discussion casually.

With regard to methodological decisions and limitations, I would like to highlight the following three aspects: Firstly, after looking at possible field sites, I decided to select hackathons that were comparatively less commercial and took place in the context of non-profit institutions. Therefore, one should also acknowledge that these hackathons are often characterized by the best intentions and aims on side of the organizers. I chose not to include strictly corporate hackathons, since their aims, suggested data and technologies (soft- and hardware) are often predefined. At the same time, I also did not investigate hackathons that were set in (techno)politically idealistic contexts, for example, organized by free software communities. This thematic focus and the restriction to two cases imply that my observations merely open up the discussion and present a perspective on certain hackathons.

Secondly, due to the open nature of my methodological approach, I did not systematically gather exact demographic information on, for instance, participants' age. While most participants were between 20 and 45 years old, I do acknowledge that more concrete demographics might be insightful. However, initial inquiries about participants' age and background rather seemed to inhibit their willingness to engage in in-depth conversations. Thirdly, I chose the two approaches

described above – being part of a developer team and being an observer – since I experienced during prior participation in hackathons that the active involvement gives me different access and insights into the digital tools used by participants. At the same time, this approach is very immersive and interviewing participants other than team members is often only possible during breaks. Moreover, during the SHD, participants might have been more hesitant to share their opinions and approaches, since they could have perceived me as member of a competing group. While I might have appeared like a ‘neutral’ external observer during *Hack4DK*, I had to approach groups successively and at different stages. This does not however imply that my approach is only partly based on digital ethnography – it merely involves different levels and types of involvement. On the one hand, ‘non-participation’ is an established element of ‘traditional’ ethnography (Gobo and Marciniak, 2011: 105ff). On the other hand, particularly the multisited studies presented in Pink et al. (2016) show that the involved researchers adjusted the type of participation to their particular field sites as well as the stage of their research process (see e.g. 131ff or 160ff). In a similar way, my approach draws continuously on digital ethnography as approach, while I have varied the individual methods used at field sites.

## Case studies and analysis: Hackathon practices

The SHD in Eindhoven (GASlab, Eindhoven University of Technology) was organized by the Mad Emergent Art Centre. SHDs are a community-run hackathon type initiated in 2010. They focus on interdisciplinary innovation and prototyping broadly related to science. Participants at the event were mainly industrial and graphic designers, artists, programmers and software developers as well as engineers. Participants worked together in groups of 2–7 people, and a jury was invited to judge their projects. It consisted of creative industry practitioners and university researchers. Prizes up to €500 were given and assigned to themes. At the end of day 2, all project teams presented their ideas and the jury allocated the prizes, sponsored by companies such as Lumens Group and Endnote.

The *Hack4DK* event (<http://hack4dk.wordpress.com>) was organized by Danish public, cultural institutions. *Hack4DK* stands for ‘hack for Denmark’. It was inspired by early hackathons using the hashtag ‘hack4EU’ initiated by the European Foundation. The initiative started in 2012 at the National Museum of Denmark (Wang, 2014). The 2014 event took place at the Copenhagen State Museum for Art. It focused on open data by public institutions such as museums and archives and invited participants to think about challenges in this sector.

At the beginning of the hackathon, speakers posed practical problems, for example, concerning the conservation of art works or audience involvement and presented heritage data sets. Most projects had a strong relation to the issues posed during the initial talks, but participants were free to choose a topic and some worked on unrelated ideas. Similar to the SHD, many participants had a professional background as programmers/developers, engineers, designers, tech-journalists or artists; only a few of them were related to open-source/open knowledge initiatives such as Wikimedia or Open Street Map. Participants received rather ‘symbolic’ prizes such as books and shirts for their projects. Moreover, the winning project was chosen based on a public vote among the participants. However, the event was used as networking platform: by pitching their ideas, some participants were hoping to receive job assignments, since this had happened after *Hack4DK* in 2013.



**Figure 1.** Hack4DK at the Statens Museum for Kunst. Source: Photo taken by the author.

### *Relevance and implications of persuasive presentations*

My first argument is that in (even mildly) competitive hackathons, strategic effort is put into persuasive presentations. The project idea needs to be convincingly presented, while the technological implementation does not necessarily have to be completely functional.

Already the spatial setup of *Hack4DK* indicates that project development practices and participants' final presentations were literally staged: the theatre-layout invited observers, and the developers were working in front of an auditorium. During the hackathon, a slide was shown which explained the event and invited viewers (see Figure 1). The project work took place in the physical setting of the museum, and the same space was later used for the presentations. As mentioned before, *Hack4DK* offered only 'symbolic' prizes, and the final projects were collectively evaluated by the participants themselves. However, participants were invited to present their projects in order to convince each other to vote for their project (and possibly receive job assignments by the event hosts). This creates the need for expertise in presentation design and persuasively pitching ideas. Next to strictly technical skills – for tasks involving coding and hardware building, for instance – rhetorical and persuasive pitching skills are also needed in the groups. This was similar in the case of the SHD; a difference here however was that monetary prizes were available and that the decision was made by a jury.

Trainer et al. (2016) differentiate between four project development phases at hackathons: (1) group *forming*, the exchange of personal information and clarification of the task, (2) individual role negotiations (*storming*), (3) process norming and decisions on the group's working style (*norming*), and (4) the group's final *performing* during which the project is created. In practice, it is likely that these phases overlap and are not strictly linear, but nevertheless I recognized similar processes during the observed hackathons. In addition to the phases outlined by Trainer et al., it

seems sensible though to add a fifth, influential phase: the *staging* of a project and the effective, persuasive pitching of an idea.

It might seem counter-intuitive to start my analysis with this aspect of hackathons, given that presentations are usually the concluding part of a hackathon. However, one has to be aware that the responsibility for preparing a presentation is often determined by groups at the very beginning. This condition is particularly relevant for my second argument on skills development and learning.

Content-wise, particularly projects' societal relevance and usefulness were important qualities. Participants extensively addressed in their presentations the purposes for which their projects could be used. For example, the group which I joined at the SHD developed a project called Connected Light. Technically, the team aimed at creating a device combining physical card gaming with virtual gaming. Its highlighted societal purpose was to allow elderly people to interact with a remote player without having to use a computer. Similarly, the project BatSense, also developed at SHD, aimed at creating a prototype for a device that translated a sensor's visual 'perceptions' into vibrations, allowing a blind person, for instance, to acquire an artificial sense. Both projects won prizes – for Social Innovation and the Best Maker Trophy.<sup>8</sup> In the context of *Hack4DK*, the societal relevance was particularly focused on usefulness in response to initially outlined challenges in museums. This also shows that the agenda of a certain host institution aims at influencing the types of projects developed. The project discussed in the following section (Raspberry Preserve) illustrates this point.

Projects involving hardware development usually presented a partly functional prototype or illustrative mock-up/simulation. In contrast, participants' programming efforts and the created code were rather invisible. Such an 'invisibility of code' might seem surprising, since programming was one of the main tasks at both events. Code sharing platforms such as *GitHub* were crucial, allowing participants to draw on a global pool of existing code, to share and document their contributions (see also Dabbish et al., 2012). In particular, during *Hack4DK* many projects were focused on web applications and consequently required programming. Participants at both events used diverse programming languages and libraries in order to realize their projects<sup>9</sup>: for example, Python, a general purpose, high-level (of abstraction) programming language, or PHP, a scripting language designed for web development (server side) which may be also used as a general purpose programming language.

However, the actual code and the process of creating it were at best mentioned very briefly in the final project presentations at both hackathons. Process shot pictures (Figure 2) were included in the presentations, but participants did not show the code in detail. Illustrative representations of coding as practice were used in the sense of technology as resource: as images hinting at a project's technical feasibility. In contrast, abstract, in-depth presentations of the code were avoided. This observation indicates that groups design their presentation in anticipation of a general audience and jury members – which also means that the overall idea pitch and feasibility arguments, rather than the software details, are decisive.

### *Efficient skills utilizations and learning limitations*

My second argument is that there is limited time for personal learning and development of new skills, since participants' expertise needs to be efficiently applied. This is particularly problematic in cases where different skill sets or levels are expressed in terms of gender.

Within project development practices during both events, all teams tended to establish a distinctive division of labour at the start of their work. In light of time pressure and competitive



**Figure 2.** Process shot showing the coding for Connected Light. Source: Photo taken by one of the participants (anonymised).

setups, group members decided to take over tasks which required skills that they already possessed. This applied to the teams to whom I spoke to at *Hack4DK* and was likewise reflected at the SHD.

Regarding the aforementioned project development phases suggested by Trainer et al. (2016) – *forming*, *norming*, *storming*, and *performing* – it is important to realize that particularly the first three stages are crucially shaped by information exchange on existing skills. Participants initially created an overview of expertise in their team and simultaneously identified necessary tasks in order to realize their idea. Subsequently, tasks were divided among the team members. Overall, there was a tendency to ensure that these were applied as efficiently as possible, in favour of the overall group performance. For example, I was mainly involved in the project's graphical work and presentation design: tasks that I am familiar with, while I am not a proficient programmer. This trend was manifested overall at the SHD and *Hack4DK*: graphic designers, journalists or public relations professionals created the presentations and pitched the idea. This also means of course that experienced programmers were less likely to take over the presentation preparation – which in turn might facilitate the aforementioned invisibility of code. When asked about this pattern, participants stated that they agreed on an allocation of tasks which seemed 'ideal', 'obvious' and 'practical' for the respective group and projects.

Participants' decisions to take over certain tasks were influenced by a sense of fairness, paired with the need for effectiveness and efficiency. Considering the competitive setup, utilising existing skills appears to be a fair, reasonable strategy. Existing skills may be considerably improved, but the potential for learning completely new skills is rather limited. The reason for this is that personal skills development did not pay off in the project presentations: projects were not assessed based on participants' personal learning curve but with regard to criteria such as innovativeness, originality and societal relevance. Trainer et al. likewise recognized that one reason why:

[...] social coding tools sometimes fell short is that not all participants were familiar with them, nor did they have the incentive to become familiar with them if they did not plan to use them beyond the hackathon. [...] Less specialized tools seem particularly important when mixing groups that do not share a common tool set. (Trainer et al., 2016: 11)

While I agree with the authors' conclusion that less specialized tools are an important deliberation in order to facilitate balanced involvement of all participants, my observations do not merely suggest that participants' did not have a personal incentive to familiarize themselves with certain tools. Instead, the competitive setup of hackathons promotes an efficient utilization of existing skills in order to contribute to the group's overall performance. Training in new skills is not merely impeded due to the limited amount of time available to hackathon participants. Initiatives such as the *Django Girls* events (which may in turn be criticized for their language of top-down empowerment, stating, for instance, 'We inspire women to fall in love with programming') or some of the sessions organized as part of the yearly *European Codeweek* (<http://codeweek.eu>) demonstrate that acquiring basic coding skills is possible within a very short time frame. However, the level of skills that can be achieved and how quickly such skills can be put to use to impresses a jury, visitors or other participants is another matter. While one might gain insights into coding and related skills that constitute a significant personal development for the individual, such learning cannot necessarily be immediately harnessed to benefit a team-based project and is hence impeded at hackathons.

This insight is particularly relevant for hackathon organizers who wish to encourage learning at their events. Likewise, it illustrates the need for critical studies regarding the learning potential, conditions and limitations of hackathons (see, for instance, Nandi and Mandernach, 2016). The described tendency is especially problematic, since different skills (levels) of hackathon participants are often gender specific. At both hackathons, female participants were less likely to engage in programming or hardware building. This issue is closely related to the IT sector and hacking communities being traditionally male-dominated domains (see, for instance, Lewis, 2015; Wajcman, 2007; Webster, 2014). In order to stay competitive, more technical tasks were taken over by participants who already possessed programming skills or hardware knowledge (e.g. see Figure 3). In many cases, female participants were responsible for design matters, in particular for creating and giving the presentation. If hackathons want to encourage participants to train in new or little developed skills and thereby contribute to addressing an existing gender bias in IT professions, organizers would need to provide incentives for personal skills development rather than foster an environment that privileges efficient skills utilization. The tendency to encourage efficiency has been countered, for example, with the recent phenomenon of *un-hackathons*:

[...] a format inspired by participatory design approaches to creatively develop technology and a rejection of high-pressure, end-product-focused hackathon practices that reward speedy coding over effective problem solving. Rather than racing to complete or compete, the un-hackathon was about collaboration, brainstorming, and better understanding [...]. (Bailey Hogarty et al., 2015)

While un-hackathons aim at overcoming the goal-oriented setup of hacking events more generally, there are also initiatives that specifically target women. For example, in the game development sector, '[t]he masculinist character of video games culture broadly, and production specifically, has prompted the creation of several initiatives to attract, support, and retain women interested in the game industry' (Harvey and Fisher, 2013: 363). Harvey and Fisher analyse the Canadian non-profit *difference engine initiative* (DEI): by involving underrepresented actors, especially women, in game development and supporting them in learning to code, this project aimed to 'diversify what kind of videogames are made' (Hand Eye Society, 2011), Harvey and Fisher also acknowledge the learning and networking benefits created by such events (see also



**Figure 3.** Tweet published by the Danish Statens Museum for Kunst. Source: <https://twitter.com/smkmuseum/status/516169653748785152>.

Harvey and Fisher, 2015). But they likewise critically assess that women's affective labour facilitated and promoted the idea of a dynamic, inclusive context for local game innovation (2013: 364ff). While the communication of values such as inclusivity and diversity was closely related to the target group selected for the DEI, one can argue that in a similar way values such as openness and flexibility are represented by the technologies used – which is the topic of the next sub-section.

### *Technology choices and utilization*

My third argument, building on the initial two observations, is that certain participants are more inclined to use technologies which are proprietary but operationally open enough. This seems

particularly relevant, since such technologies are staged as potent enablers of innovation, especially during the final presentations.

While presenting their projects, participants particularly drew attention to the hardware that enabled the realization of their ideas. Two devices were most frequently used and presented in this context: Raspberry Pi and Arduino (Uno). It seems insightful to look into the chosen hardware, since the presentations implicitly communicated that these tools are well-suited for innovation practices, as a '[...] resource not only to experience content, but to modify it and to participate in it' (Postigo, 2012: 176).

## Raspberry Pi

The Raspberry Pi is a microcomputer that is used for a wide range of single board computing projects. The product was developed by the Raspberry Pi Foundation, a registered UK charity since May 2009. While the device involves OSH units such as openly documented graphics hardware, main parts such as the CPU/chip design are closed source. In an interview, Eben Upton – co-founder of the Raspberry Pi Foundation – explained the reasons for the proprietary hardware approach:

Raspberry Pi is not open hardware, and there is no plan currently to release the board design. . . . While we're supportive of the open hardware movement, we don't believe that releasing designs for very high-technology, hard-to-manufacture products like the Pi brings significant direct benefits to end users. (Brodin, 2014)

Regarding the relevance of technological openness, the Raspberry Pi is described as 'open-source enough' since it enables the creativity and innovation capacity of users/individual customers. The issue addressed with this device is what can be learnt *with* rather than *from* the hardware. The ways in which Raspberry Pi was used and staged during both events correspond with these aims: as a 'productive' resource facilitating the development process.

During *Hack4DK*, a project called Raspberry Preserve showed how the hardware may be put to use: it was presented as tool for art conservation, measuring temperature and humidity in a space. The data would be documented online and warning emails sent once the measurements exceeded a certain value.<sup>10</sup> The project is particularly interesting since it shows on the one hand why participants might choose certain technologies; on the other hand, it illustrates how hardware is staged and presented.

The project was realized in different ways by two participants. While using the same hardware (Raspberry Pi and a sensor for humidity/temperature detection), they employed different software approaches. One participant created the required software 'from scratch' and programmed the necessary code in Python. He was aware that there were easier options, but he wanted to realize the project in a way that made it independent of external services or providers. The other participant drew on a Google application programming interface. The main reason, he explained, was that this year – after having attended *Hack4DK* for the past 2 years – he wanted to create something he was actually able to complete. To a large extent, the time pressure during hackathons, the final presentations and competitive setup encourage result-focused practices and technology choices, since the process of personal learning is often less acknowledged and rewarded (by, for instance, peer and audience acknowledgement, networking or prizes). As this case illustrates though, such a 'hacker pragmatism' might not apply to all participants.

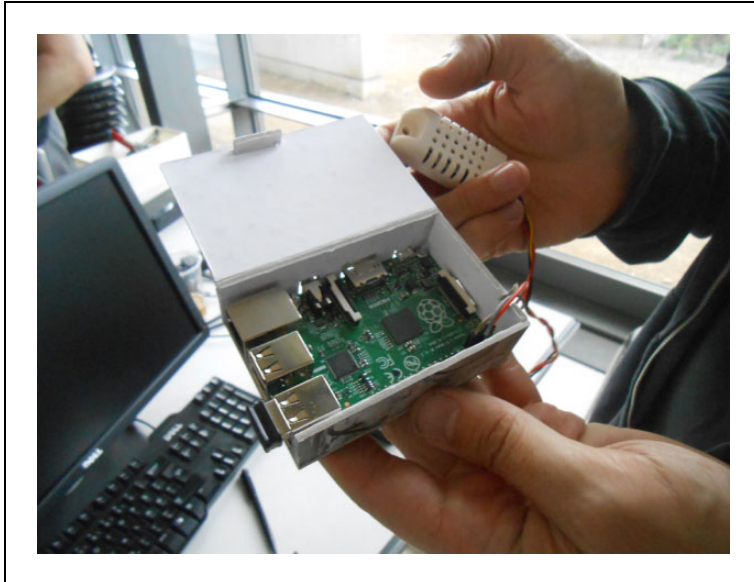




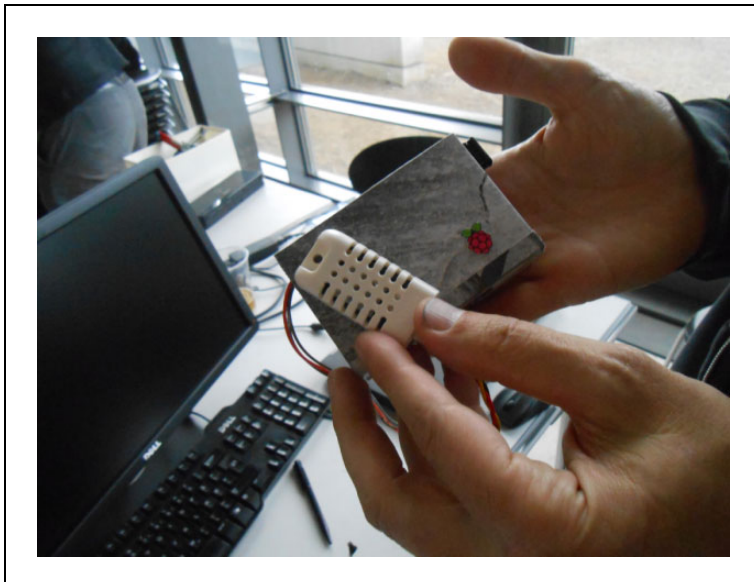
**Figure 4.** Final presentation of Raspberry Preserve at Hack4DK. *Source:* Photo taken by the author.

In both cases, however, the Raspberry Pi logo was prominently featured in the presentations and staged as ‘the hardware behind the projects’. It was shown in presentation slides (Figure 4) and also added separately onto the box used for the demonstration of the prototype (Figures 5 and 6). Moreover, the Raspberry Pi was already featured prominently as part of the prototype name. After the hackathon, the project’s further development was circulated via Twitter (Figure 7). Thus, the technology as such was mediated in multiple ways: it was not simply presented as a technological prototype but was literally boxed as a material digital object (Figures 5 and 6) and linguistically framed as part of the project’s title. In terms of technology as resource, the Raspberry Pi was shown as a potent device supporting techno-creative practices. Only a closer look at the hardware conditions reveals that in the sense of technology as opportunity, developer’s interactions are limited to the software level. Also, on the surface – with regard to the presented hardware which was the same for both approaches to the *Raspberry Preserve* – the different software approaches taken were not apparent.

Besides these actual implementations, the Raspberry Pi was also used as indicator of a project’s realistic implementation – as ‘signifier of its feasibility’. During the SHD, the hardware

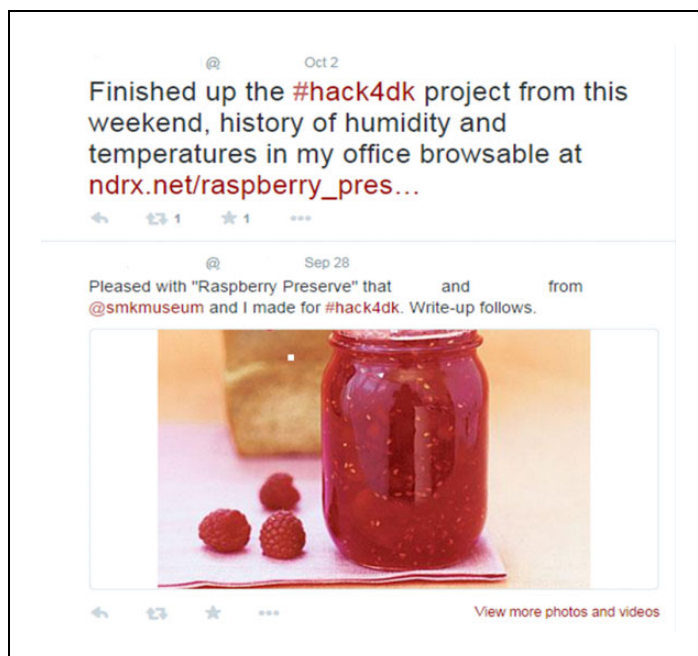


**Figure 5.** Inside of the Raspberry Preserve prototype. Source: Photo taken by the author.

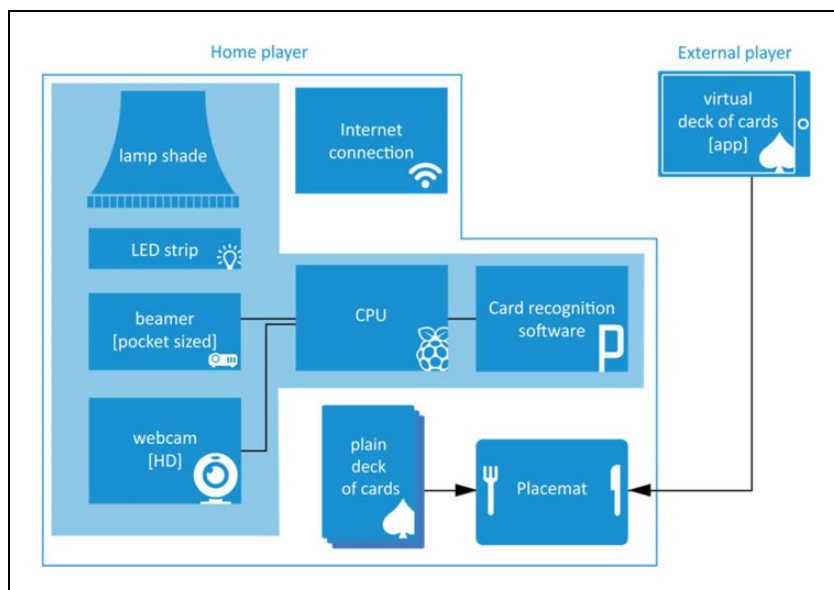


**Figure 6.** The Raspberry Pi logo, added to the box of the Raspberry Preserve prototype. Source: Photo taken by the author.

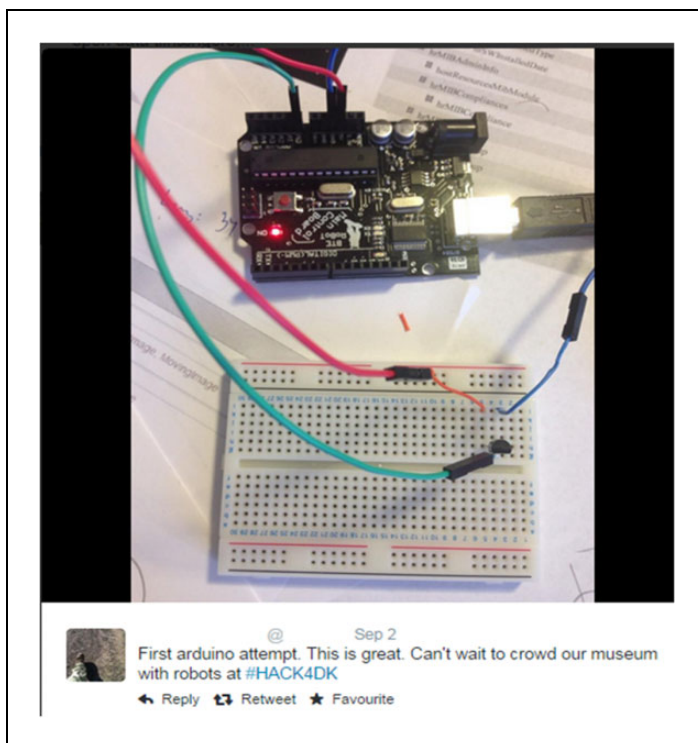
implementation of Connected Light had to be simulated and was only suggested in theory. However, the schematic overview shown in the final presentation concretely suggested using a Raspberry Pi (see Figure 8), using the hardware's image as a potent enabler of digital creativity and



**Figure 7.** Tweet on the finalization of Raspberry Preserve. Source: [https://twitter.com/fairchild\\_dk/status/517681970634326017](https://twitter.com/fairchild_dk/status/517681970634326017).



**Figure 8.** Schematic overview of Connected Light presented at the Science-Hack-Day. Source: Graphic created by one of the participants (anonymised).



**Figure 9.** Tweet published before the start of Hack4DK 2014. Source: <https://twitter.com/wulfthor/status/506761415387275264>.

realistic solutions. In this context, the technology was not explained in detail, but the Pi was presented rather self-evidently as a commonly known solution for such purposes. Here, the participants built on a pre-established image of the technology and mobilized positive connotations related to technological openness.<sup>11</sup>

### Arduino (Uno)

Besides the Raspberry Pi, Arduino hardware was relevant to both events, particularly at the SHD: Arduino is a small microcontroller built on a printed circuit board, which supports the development of interactive projects. It can be connected to sensors and motors, allowing interaction with its physical environment. One might find some debate about whether Arduino is in fact open source (regarding up to which level), however, it can be seen as an OSH as well as open-source software project since the designs are available under copyleft licenses.<sup>12</sup> Different boards are available, but I encountered mainly the *Arduino Uno*.

Before the start of *Hack4DK*, tweets about projects involving Arduino already announced its use (Figure 9). On-site, the device was not that visible though. In comparison, during the SHD more hardware innovations were presented and Arduino was used for several projects: it was not only presented graphically in presentations or online, but also in prototype demonstrations. Projects involving Arduino included for example the aforementioned ‘Batsense’, a wearable for

visually handicapped users. While the prototype was presented on-site, the presentation likewise showed visualizations of the board. Another example was 'VR Wind': 'a ventilator that can be attached to an Oculus Rift [...] made of a desktop pc's casefan, Arduino, LEGO pieces, ducttape and some lines of C++' (<http://nl.linkedin.com/in/jurgenvandervlist>). Only in one case had the board been built individually, in advance to the event (<http://renwired.com/e-trabant/kleinste-e-trabant>). In all these cases, Arduino as resource indicates that open hardware and software solutions enable digital creativity and rapid prototyping. It demonstrates what kind of civic innovations are possible when enabling certain modes of cultural production through technological openness and information transparency. One needs to keep in mind though that the openness of the hardware was in fact rarely used.

Both hackathons indicate that the time pressure and competitiveness of such events encourage pragmatic decision-making and interactions. The utilized technologies need to support straightforward, rapid development processes, and the same goes for participants' skills. This observation provides further support to Pink et al.'s emphasis on processes of co-constitution (2016: 152) – echoing, of course, the notion of co-construction which has long been crucial to science and technology studies as well as other disciplines critical of techno-deterministic theories. Here, participants' attitudes, project development decisions, technology choices and the digital material event environment yield intricate interplay and event dynamics. While proprietary options such as application programming interface (APIs) lock projects into a position depending on the provider, they may allow participants to realize functional, creative prototypes more easily and within a short time. West (2003: 1382) described such technological options as 'partly-open strategy' aimed at 'attracting user-programmers': 'Such strategies presume that there is an intermediate level of disclosure and granting of rights (between fully open and fully proprietary extremes) that will be valuable to users without compromising the IPR owners' competitive concerns' (West, 2003: 1382). Various reasons may contribute to participants' inclination to draw on such partly open options: during hackathons, it seems to be mainly a question of technologies' competitiveness and efficiency.

Participants' technology choices seemed related to two crucial factors: Does the technology allow for its adjustment to the project's purposes; and can this adjustment be realized quickly and easily? While open-source options may meet the first criterion, they often come along with a high appropriation threshold in terms of required knowledge, skills and time. Moreover, factors such as technological independence are not assessed during most hacka-thons and such a criterion does therefore not act as potential encouragement for the use of open-source software. Zooming out of the local hackathon spaces, these observations also indicate the development that closed-source software has adapted features of open-source software. West (2003: 1259) described this as a trend towards 'hybrid standards strategies [which] reflect the competing imperatives for adoption and appropriability'. While appropriability refers to approaches which ensure that a company will receive returns on their innovation (e.g. through licensing), adoption denotes the actual acceptance and usage of a product (which is in turn essential for a product's economic success). A 'partly open' strategy mediates between these factors. However, while a developer might be practically able to use an API in a similar way to an open-source code, the legal implications and dependences are crucially different. Such technologies imply that certain modes of cultural production can be achieved even with closed-source software. At the same time, they enforce usage conditions that are based on a hierarchical relationship between individual developers and corporations such as Google Inc. providing the API. In the sense of

technology as opportunity, they create a hierarchical relationship between consumers and content (Postigo, 2012: 177) by enforcing technological and legal dependencies. As linguistic resource, staged at hackathons, they signify that digital creativity may thrive even under proprietary conditions and information/code non-transparency.

## Conclusions: Persuasive, efficient, open enough?

Competitive hackathons overall encourage participants to be pragmatic about their decisions and involvement. Project development practices are decisively co-constituted by the need to efficiently utilize team members' skills and available technologies. Firstly, since convincing, final presentations and persuasively pitching an idea increase the likeliness that a group might be selected as winner, significant efforts are invested in the final phase of *staging* the project. This encourages participants to create narratives around a project's societal relevance, while detailed reflections on the code may be less prominent. Secondly, when working in teams, participants usually start their work by deciding on a division of labour within the group. In order to stay competitive, they efficiently allocate specialized tasks to group members who have experience in a certain subject area already. In particular, participants joining with comparatively little knowledge in programming or hardware building are unlikely to develop or train in such digital skills during a competitive hackathon. Instead, to support the group's competitiveness, they are more likely to apply their existing skills for familiar tasks. Such a tendency is especially problematic, since different skills (and skill levels) at hackathons are often expressed in terms of gender. Practically speaking, this is relevant to organizers who wish to facilitate participants developing new skills<sup>13</sup> and to tackle a gender-related IT skills bias. It has also been highlighted in research on game development events that the underrepresentation of certain actors in the actual development process may impede the overall inventiveness of techno-creative initiatives (see, for instance, Harvey and Fisher, 2013, 2015).<sup>14</sup>

Thirdly, competitive hackathons tend to promote the use of operationally 'open-enough', but proprietary technology such as 'open APIs'. Proprietary options were accepted as long as they enabled quick and straightforward development processes. Criteria such as projects' sustainable technological independence, as enabled by open-source-software, were prone to being neglected by participants, since these aspects were not rewarded in terms of projects' evaluation. For many participants, technologies merely needed to be open enough – so that they were able to interact with them in a productive way.

This 'hacker pragmatism' has also been highlighted in other developer communities and contexts via the notion of 'open source pragmatist' (Raymond, 2005; Yeats, 2007). Workflows and project development at the investigated hackathons indicate that initial advantages of open-source software/OSH are often countered by an increasing openness of proprietary options. Open APIs are an illustrative example: while the actual software may be in fact proprietary, they offer the opportunity to interact and access a software application. As Lyman (2012) argues in a controversial online discussion, we may have 'reached a point when non-open source software is often "open enough") for certain audiences:

APIs are where a lot of the action is and the same kind of buzz that fuelled open source. Open source still has its advantages and benefits that cannot be matched by APIs, but OSS supporters and vendors need to be aware the closed competition is also evolving and getting smarter – and more open. (Lyman, 2012)

It is important to realize here that in the sense of technology as opportunity, proprietary options may allow for interaction possibilities similar to those enabled by open-source tools. In turn though, they enforce production conditions defined by hierarchical relations between individual users and corporations as well as technological and legal limitations. This is particularly relevant since such proprietary technologies are staged during hackathons in the sense of technology as resource: as tools which enable civic, technological innovation. During competitive events, the aforementioned hacker pragmatism may further foster the use of technologies which do not reflect values such as freedom of information, technological independence, and information transparency. As I mentioned initially, of course, the outlined arguments will most likely only apply to hackathons with competitive setups and which are not inherently politically/idealistically motivated – in contrast to initiatives such as the GNU 30th anniversary hackathon.

Competitive setups, prizes, peer pressure and tight schedules at hackathons encourage participants to go for quick fixes, often enabled by open-source alike tools such as proprietary ‘open’ APIs. These hacking events likewise encourage groups to focus on persuasive project presentations rather than the development process and participants’ individual learning curve. Since the threshold in terms of necessary skills and time for using open-source options is often higher than for using proprietary solutions, participants tend to choose technology based on the criterion that it is ‘open enough’, that is, allowing for sufficient creative interaction. This dynamic fosters development practices during which participants’ personal skills development and projects’ sustainable technological independence are secondary.

Drawing on digital ethnography for an investigation of two case studies, this article illustrates how hackathons as digital-material environments co-constitute participants’ interaction and technology use. It shows that time pressure and (even mildly) competitive inducements may impede inclusive, instructive approaches: during events frequented by audiences with different IT skill levels, these conditions may inhibit those participants with lower skill levels to train in even basic expertise in, for instance, coding or electronics. Moreover, they encourage project development processes which are very focused on efficient group work and the final ‘staging’ of results. In light of some of the broader tendencies in hacking cultures such as the notion of ‘open-source pragmatist’, one may be inclined to ask to what extent these empirical observations may be indicative of broader developments.

In particular, issues related to creativity and inclusivity are noteworthy here. Regarding creativity, the term ‘hacking’ has traditionally referred to the process of solving an issue in a smart, possibly unexpected way. In contrast, the type of hacking which is now dominant at many hackathons indicates rather result-oriented approaches to innovation and how it is ultimately presented. Hartmann et al. (2008) suggest the notion of ‘opportunistic design’ in order to denote quickly implemented, experimental hacks and prototypes at events such as hackathons. They use this term to describe, for instance, approaches such as ‘copying and pasting source code from public online forums into one’s own scripts’ or “‘Frankensteining’ software and hardware artifacts together by joining them with physical and digital hot glue and duct tape’ (p. 46). The authors are not critical of these practices per se: they do note, however, that they usually ‘happen on short timelines’ (p. 52). While it should not be simply assumed that this kind of creativity is necessarily less preferable to others, it seems worth investigating how dominant it may be in contemporary hacking practices and what this implies for notions of technological creativity. Regarding diversity, this is not yet a strong feature of hacking culture/s overall – mostly evident in terms of gender, but also with regard to ethnicity (Adam, 2003; Fox et al., 2015; Henry, 2014; Jordan, 2008: 125ff.). Efforts to counter this and enhance inclusivity in civic developer contexts are evident in the

development of feminist hackerspaces (Toupin, 2014), as well as feminist hackathons (D'Ignazio et al., 2016). That this is a well-known issue, however, only stresses the need to devote more attention to investigating gender- and social diversity-related issues in this field. In particular, we need insights into factors which may inhibit or facilitate the involvement of underrepresented social groups in techno-creative initiatives and how likely they are to engage in particular tasks. While this article contributes to the conceptual debate regarding our scholarly understanding of hacking and digital creativity, it also highlights the need to develop inclusive approaches to promoting IT skills acquisition.

## Acknowledgements

The author would like to thank the two anonymous reviewers for their constructive, valuable feedback on this article, the editors for their support, and Karin Wenz for involving her in the *NWO: KIEM (Netherlands Organisation for Scientific Research: Knowledge Innovation Mapping) project "Hacking Heritage"* which inspired the research for this paper.

## Notes

1. See <https://www.gnu.org/gnu30/celebration>.
2. See <https://www.facebook.com/hackathon>.
3. In her article, Coleman (2004) describes how the 'informal politics' of FOSS emerge not only despite but in fact due to an explicit 'political agnosticism' and the circulation of technologies representing values such as free speech and the questioning conventional intellectual property rights.
4. See <http://opensource.org/definition>.
5. See <http://www.opendesign.org>.
6. While more 'traditional' ethnographic studies often present results in a 'thick description', the following analysis does not focus on such an atmospheric depiction of my results but rather – in light of the research question – highlights the connection between contextual factors, participants' interaction and decisions.
7. With 'informal interviews', I am referring to conversations which were informal in the sense that they took place on-site while participants were working on their projects or having lunch/dinner. At the beginning of these conversations, I informed all participants about my interest as a researcher. I did not tape these discussions in order to avoid any interference or confusion that might arise due to using recording technology.
8. In the case of the group I was involved in, the prize was divided among my fellow group members.
9. A list of software used during *Hack4DK* can be found on <https://github.com/NationalMuseumofDenmark>.
10. See [http://ndrx.net/raspberry\\_preserve](http://ndrx.net/raspberry_preserve).
11. In this context, it may be considered somewhat ironic that 'openness' is likewise one of the main principles of digital ethnography according to Pink et al. (2016: 11/12).
12. See <http://arduino.cc/en>.
13. See also Nandi and Mandernach (2016) on hackathons as informal learning environment.
14. The authors likewise emphasize that a focus on marginalized groups in indie game development has not only idealistic reasons, but might be entangled with commercial interests.

## References

Acid Phreak (1990) Is computer hacking a crime? *Harpers Magazine*. Available at: <http://www.textfiles.com/news/hackers.txt> (accessed 10 May 2016).



- Adam A (2003) Hacking into hacking: Gender and the hacker phenomenon. *ACM SIGCAS Computers and Society* 33(4): 3.
- Aungst TD (2015) Using a hackathon for interprofessional health education opportunities. *Journal of Medical Systems* 39(5): 1–2.
- Aviram M (1999) JavaOne's palm-sized winner. *Javaworld*. Available at: <http://www.javaworld.com/article/2076473/mobile-java/javaone-s-palm-sized-winner.html> (accessed 10 May 2016).
- Bailey Hogarty S, Winesmith K, and Hrudka M (2015) Art + data: Building the SFMOMA collection API. In: *Museums and the Web 2015*. Available at: <http://mw2015.museumsandtheweb.com/paper/art-data-building-the-sfmoma-collection-api> (accessed 10 May 2016).
- Bergquist M, Ljungberg J, and Rolandsson B (2011) A historical account of the value of free and open source software: From software commune to commercial commons. In: Hissam S, Russo B, de Mendonça Neto M, et al. (eds), *Open Source Systems: Grounding Research*. Berlin/Heidelberg: Springer, pp. 196–207.
- Briscoe G and Mulligan C (2014) Digital innovation: The hackathon phenomenon. Available at: <http://www.creativeworkslondon.org.uk/wp-content/uploads/2013/11/Digital-Innovation-The-Hackathon-Phenomenon1.pdf> (accessed 10 May 2016).
- Brodkin J (2014) Raspberry Pi marks 2nd birthday with plan for open source graphics driver. In: *Ars Technica*. Available at: <http://arstechnica.com/informationtechnology/2014/02/raspberry-pi-marks-2nd-birthday-with-plan-for-open-source-graphics-driver/> (accessed 10 May 2016).
- Coleman G (2004) The political agnosticism of free and open source software and the inadvertent politics of contrast. *Anthropological Quarterly* 77(3): 507–519.
- Coleman G (2009) Code is speech: Legal tinkering, expertise, and protest among free and open source software developers. *Cultural Anthropology* 24(3): 420–454.
- Coleman G (2012) *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press.
- Coleman G (2014) *Hacker, Hoaxer, Whistleblower, Spy: The Story of Anonymous*. New York: Verso.
- Coleman G and Golub A (2008) Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory* 8(3): 255–278.
- Couldry N (2012) *Media, Society, World: Social Theory and Digital Media Practice*. London: Polity.
- Dabbish L, Stuart C, Tsay J, et al. (2012) Social coding in GitHub: Transparency and collaboration in an open software repository. In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, Seattle, Washington, USA, 11–15 February 2012, pp. 1277–1286.
- D'Ignazio C, Hope A, Metral A, et al. (2016) Towards a feminist hackathon: The “make the breast pump not suck!” hackathon. *Journal of Peer Production*. Available at: <http://peerproduction.net/issues/issue-8-feminism-and-unhacking/peer-reviewed-papers/towards-a-feminist-hackathon-the-make-the-breast-pump-not-suck> (accessed 22 November 2016).
- Dominguez D, Beaulieu A, Estalella A, et al. (2007) Virtual ethnography. *Forum Qualitative Social Research* 8(3). Available at: <http://www.qualitative-research.net/index.php/fqs/article/view/274/601> (accessed 10 May 2016).
- Folz R (2015) Selling Out and the Death of Hacker Culture. Available at: <https://medium.com/@folz/selling-out-and-the-death-of-hacker-culture-fec1f101b138> (accessed 10 May 2016).
- Fox S, Ulgado RR, and Rosner D (2015) Hacking culture, not devices: Access and recognition in feminist hackerspaces. In: *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*, pp. 56–68. New York: ACM.
- Gatson SN and Zweerink A (2004) Ethnography online: ‘Natives’ practising and inscribing community. *Qualitative Research* 4(2): 179–200.
- Gobo G and Marciniak LT (2011) Ethnography. In: Silvermann D (ed), *Qualitative Research*. London: SAGE, pp. 103–120.
- Gregg M (2015) Hack for good: Speculative labour, app development and the burden of austerity. *The Fibreculture Journal*. Available at: <http://twentyfive.fibreculturejournal.org/fcj-186-hack-for-good-speculative-labour-app-development-and-the-burden-of-austerity> (accessed 15 November 2016).

- Hand Eye Society (2011) The difference engine initiative. Available at: <http://handeyesociety.com/difference-engine-initiative> (accessed 15 November 2016).
- Hartmann B, Doorley S, and Klemmer S (2008) Hacking, mashing, gluing: Understanding opportunistic design. *IEEE Pervasive Computing* 7(3): 46–54. Available at: <http://hci.stanford.edu/publications/2008/hackingmashinggluing.pdf> (accessed 22 November 2016).
- Harvey A and Fisher S (2013) Making a name in games. *Information, Communication and Society* 16(3): 362–380.
- Harvey A and Fisher S (2015) ‘Everyone can make games!’: The post-feminist context of women in digital game production. *Feminist Media Studies* 15(4): 576–592.
- Henry L (2014) The rise of feminist hackerspaces and how to make your own. Available at: <https://modelviewculture.com/pieces/the-rise-of-feminist-hackerspaces-and-how-to-make-your-own> (accessed 27 March 2017).
- Himanen P (2010) *The Hacker Ethic*. New York: Random House.
- Hine C (2000) *Virtual Ethnography*. London: SAGE.
- Hine C (2008) Virtual ethnography: Modes, varieties, affordances. In: Fielding NG, Lee RM, and Blank G (eds) *The SAGE Handbook of Online Research Methods*. London: SAGE, pp. 257–270.
- Hippel E and Krogh G (2003) Open source software and the ‘private-collective’ innovation model: Issues for organization science. *Organization Science* 14(2): 209–223.
- Irani L (2015) Hackathons and the making of entrepreneurial citizenship. *Science, Technology and Human Values* 40(5): 799–824.
- Johnson P and Robinson P (2014) Civic hackathons: Innovation, procurement, or civic engagement? *Review of Policy Research* 31(4): 349–357.
- Jordan T (2008) *Hacking: Digital Media and Technological Determinism*. Cambridge: Polity Press.
- Jordan T (2016) A genealogy of hacking. *Convergence: The International Journal of Research into New Media Technologies*. Epub ahead of print 6 April 2016. DOI: 10.1177/1354856516640710.
- Jordan T and Taylor P (2004) *Hactivism and Cyberwars. Rebels with a Cause?* Abingdon: Routledge.
- Kaltenbrunner M and Ehtler F (2014) TUIO hackathon. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, Dresden, Germany, 16–19 November 2014, pp. 503–505.
- Kelty CM (2008) *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press.
- Kogut B and Metiu A (2001) Open-source software development and distributed innovation. *Oxford Review of Economic Policy* 17(2): 248–264.
- Levy S (1996) [1984] Hackers: Heroes of the computer revolution. Available at: <http://www.gutenberg.org/ebooks/729> (accessed 10 May 2016).
- Lewis J (2015) Barriers to women’s involvement in hackerspaces and makerspaces. In: *Access as spaces*. Available at: <http://access-space.org/wp-content/uploads/2015/10/Barriers-to-womens-involvement-in-hackerspaces-and-makerspaces.pdf> (accessed 10 May 2016).
- Lin YW (2007) Hacker culture and the FLOSS innovation. In: St. Amant K and Still B (eds), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. Hershey and New York: Information Science Reference, pp. 34–46.
- Lyman J (2012) Open APIs Are the New Open Source. In: *Linuxinsider*. Available at: <http://www.linuxinsider.com/story/74419.html> (accessed 10 May 2016).
- McLellan A, Dubin RA, and Jing Q (2012) The wasp system: An open source environment for managing and analyzing genomic data. *Genomics* 100(6): 345–351.
- Miller D and Slater D (2001) *The Internet: An Ethnographic Approach*. London: Bloomsbury.
- Moyer PR and Schultz EE (1996) A systematic methodology for firewall penetration testing. *Network Security* 1(3): 11–18.
- Munro D (2015) Hosting hackathons: A tool in retaining students with beneficial side effects. *Journal of Computing Sciences in Colleges* 30(5): 46–51.

- Nandi A and Mandernach M (2016) Hackathons as an informal learning platform. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, Memphis, Tennessee, USA, 02–05 March 2016, pp. 346–351.
- Nissenbaum H (2004) Hackers and the contested ontology of cyberspace. *New Media and Society* 6(2): 195–217.
- O’Boyle N, Banck M, and James CA (2011) Open babel: An open chemical toolbox. *Cheminf* 3: 33.
- OpenBSD n.d. Hackathons. Available at: <http://www.openbsd.org/hackathons.html> (accessed 10 May 2016).
- Pearce JM (2012) The case for open source appropriate technology. *Environment, Development and Sustainability* 14(3): 425–431.
- Pink S, Horst H, Postill J, et al. (2016) *Digital ethnography: Principles and practice*. London: SAGE.
- Porway J (2013) You can’t just hack your way to social change. Available at: <https://hbr.org/2013/03/you-cant-just-hack-your-way-to> (accessed 10 May 2016).
- Postigo H (2012) *The Digital Rights Movement: The Role of Technology in Subverting Digital Copyright*. Cambridge: The MIT Press.
- Powell A (2012) Democratizing production through open source knowledge: From open software to open hardware. *Media, Culture and Society* 34(6): 691–708.
- Quilez I and Pol J (2014) Shader toy hackathon. *SIGGRAPH 2014 Studio*. Available at: <http://dl.acm.org/citation.cfm?id=2656312> (accessed 10 May 2016).
- Ray B, et al. (2014) A large scale study of programming languages and code quality in github. In: *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. Available at: [http://macbeth.cs.ucdavis.edu/lang\\_study.pdf](http://macbeth.cs.ucdavis.edu/lang_study.pdf) (accessed 10 May 2016).
- Raymond E (2005) *Homesteading the Noosphere*. Available at: <http://firstmonday.org/ojs/index.php/fm/article/view/1474/1389> (accessed 10 May 2016).
- Rosell B, et al. (2014) Unleashing innovation through internal hackathons. In: *Innovations in Technology Conference*. Available at: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6870526> (accessed 10 May 2016).
- Söderberg J (2008) *Hacking Capitalism: The Free and Open Source Software Movement*. London: Routledge.
- Stallman R (2007) Why open source misses the point of free software. Available at: <http://www.gnu.org/philosophy/open-source-misses-the-point.en.html> (accessed 10 May 2016).
- Suchman L, Trigg R, and Blomberg J (2002) Working artefacts: Ethnomethods of the prototype. *The British Journal of Sociology* 53(2): 163–179.
- Toupin S (2014) Feminist hackerspaces: The synthesis of feminist and hacker cultures. *The Journal of Peer Production*. Available at: <http://peerproduction.net/issues/issue-5-shared-machine-shops/peer-reviewed-articles/feminist-hackerspaces-the-synthesis-of-feminist-and-hacker-cultures> (accessed 22 November 2016).
- Trainer EH, Kalyanasundaram A, and Chaihirunkarn C (2016) How to hackathon: Socio-technical tradeoffs in brief, intensive collocation. In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. New York: Available at: <http://www.cs.cmu.edu/~etrainer/papers/trainer-cscw2016-hackathons.pdf> (accessed 10 May 2016).
- Underberg NM and Zorn E (2013) *Digital Ethnography: Anthropology, Narrative, and New Media*. Austin: University of Texas Press.
- Wajcman J (2007) From women and technology to gendered technoscience. *Information, Community and Society* 10(3): 287–298.
- Wang J (2014) Digital cultural heritage: Long perspectives and sustainability. *Sharing is Caring. Openness and Sharing in the Cultural Heritage Sector*. Available at: <http://www.smk.dk/en/explore-the-art/free-download-of-artworks/sharing-is-caring/jacob-r-wang> (accessed 10 May 2016).
- Wark MK (2013) Considerations on a hacker manifesto. In: Scholz T (ed), *Digital Labor: The Internet as Playground and Factory*. New York: Routledge, pp. 69–76.
- Webster J (2014) *Shaping Women’s Work: Gender, Employment and Information Technology*. London and New York: Routledge.

- West J (2003) How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32(7): 1259–1285.
- Wiggins A and Winter S (2014) OData Hackathon @ CSCW 2014: Online communities data hackathon. In: *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 317–318, New York: ACM.
- Yeats D (2007) Morality and pragmatism in free software and open source. In: St. Amant K and Still B (eds), *Handbook of Research on Open Source Software: Technological, Economic and Social Perspectives*. Hershey and New York: Information Science Reference, pp. 23–33.
- Zhao L and Elbaum S (2003) Quality assurance under the open source development model. *Journal of Systems and Software* 66(1): 65–75.

### Author biography

**Annika Richterich** is an Assistant Professor in digital culture at Maastricht University's Faculty of Arts and Social Sciences (NL). She studied sociology at University of Auckland (NZ) and media economics at University of Siegen (GER). Together with Karin Wenz (PI) she conducted a research project on hackathons (June 2014–June 2015). She is the associate editor of the *Digital Culture & Society* journal and currently co-edits an issue on “Making and Hacking”. Moreover, she has published on the utilization of ‘big data’ and critically evaluated its ethical implications for academic research.